IJCSN International Journal of Computer Science and Network, Volume 3, Issue 5, October 2014
ISSN (Online) : 2277-5420 **www.IJCSN.org**
Impact Factor: 0.274

A Survey on Intrusion Detection Systems for Android Smartphones

[1] Chani Jindal, [2] Mukti Chowkwale, [3] Rohan Shethia, [4] Sohail Ahmed Shaikh

[1, 2, 3, 4] Student, Department of Computer Engineering, MIT College of Engineering
Pune - 411038, Maharashtra, India

Abstract - Smartphones have become a popular and an imperative form of mobile computing devices. With the proliferation of smartphones however, the security threats have correspondingly increased. Although, some form of security mechanisms like authentication and encryption have been provided on platforms such as Android and iOS, these alone cannot mitigate all the forms of threats. Malwares for smartphones is also on the rise and pose a grave security threat. Hence the need for an intrusion detection system for smartphones has become immensely important. This paper aims to discuss the current trends in intrusion detection mechanisms for smartphones. The important features that such a system should have are network traffic monitoring, classification of the packets and reporting to the user in the event of an attack. Further, we expound on the types of an IDS and explore its features in detail. The open issues concerning the implementation of IDS have also been discussed.

Keywords - *Intrusion Detection Systems, Android, Information Security, Machine Learning.*

1. Introduction

During recent years, the share of smartphones in overall handheld mobile communication device sales has drastically increased. Among them, Android phones dominate the market. In Q2 2014, 84.7% of all devices sold were Android devices, followed by Apple's iOS (11.7%), according to IDC analysis. [16] The openness and programmability of Android makes smartphones more vulnerable to various malicious attacks, such as Trojan horses, worms, mobile botnets, and so on. These malicious attacks may result inserious damage to mobile phone users, including system corruption, individual privacy divulgence, malicious fee deduction, etc. [14] Although mobile device OSs have some integrated security mechanisms, the lack of intelligent protection mechanisms leads to a situation where a number of new security threats appear every day to the mobile environment. Knowing the increasing risk of mobile malware, designing a secure mobile device that protects user's privacy is still regarded a very challenging task. Taking all the above into account, we can see that more intelligent and sophisticated security controls, such as IDSs for mobile devices are deemed necessary. Such a mechanism would enable the monitoring of the device at all times, thus greatly contributing to the issue of user post-authentication (by means of continuous authentication). This means that such an IDS could constantly track the behavior of (a) the user when interacting with the device, (b) the software state of the installed on the device, and (c) the status of the running services. Intrusion detection for smartphones is a complex and difficult task mainly due to their dynamic nature, highly constrained nodes and the lack of central monitoring points. Conventional IDSs are not easily applied to them. New approaches need to be developed or else existing approaches need to be adapted for them.

2. Intrusion Detection Systems

An intrusion detection system is introduced to in order to detect intrusions when other protective measures fail, by passively monitoring system events and looking for security related problems. The intrusions that these systems analyze are defined as attempts to compromise confidential information, integrity or to bypass security mechanisms of the host or the network. [1] An intrusion detection system monitors the actions in the environment, and decides whether these actions constitute an attack legal use of the environment.

An intrusion detection system should address the following issues, regardless of what mechanism it is based on:

- It should support the security policies and the business operations of the organization.
- It should run continuously without human supervision.

- It must be fault tolerant, i.e., it must survive a system crash and not have its knowledge base rebuilt at restart.
- It must resist subversion.
- It must impose minimal overhead on the system.
- It must observe deviations from normal activity.
- It must be easily customised to the system in question.
- It must cope with changing system behavior over time as new applications are being added.
- It must be difficult to fool even with full knowledge of internal workings by attackers. [18, 20]

2.1. Classification of Intrusion Detection Systems

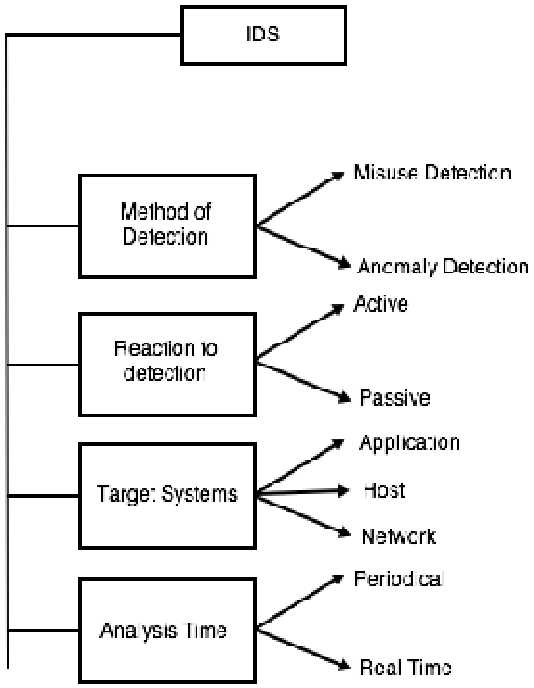

Fig1. Classification of IDS [12]

Modern IDSs have the ability to detect and respond in real time. These systems are hence called active IDS as they aim to not only monitor behavior of users on networks or systems but also prevent intrusions. The application of a countermeasure is made safer by adding the capability of reverting to a former state in the event of an intrusion or abnormality. Majority of the existing intrusion detection systems provide passive response. Among 20 IDS evaluated by Axelsson [13], 17 systems supported passive response while only 3 systems were designed to mitigate the damage or harm the attacker.

Generally, the IDS/IDPS can be put into two categories, based on the levels of a system at which it collects information. They are network level and host level. A *network-based* (NIDS) or *centralized IDS* (CIDS) examines the traffic by using a set of hardware or software sensors placed at various points in the network. A *host-based* (HIDS) monitors events occurring at a single host system two types of sources: OS Method Call trails and System Logs. Network-based systems make assumptions about network pathologies like packet fragmentation and suffer from exhaustion of resources when attack-state information must be maintained over a long period of time. However, despite these deficiencies, they are preferred as they are easy to deploy and have little or no impact on the system's performance. Host-based systems monitor specific applications in ways network-based systems cannot. They detect intrusive activities that do not create externally observable behavior. However, as they consume resources on the host, they affect performance substantially. [2]

Presently, there are two main approaches for distinguishing the occurrence of intrusions: misuse detection and anomaly detection. *Misuse-based* or *Signature-based* usually distinguishes abnormal behavior by matching it against pre-defined patterns of known attacks. *Anomaly-based* or *Behavior-based* [3] intrusion detection, represents normal behavior and attempts to identify anomaly patterns of activities that vary from the the defined profiles. Further, anomaly detection is broadly classified into: statistical-based, knowledge-based and machine learning-based. In the statistical method the behavior is profiled from a random viewpoint. Knowledge-based systems capture the claimed behavior. Finally machine-learning techniques learns to distinguish complex patterns and make intelligent decisions. Anomaly detection requires normal data while building profiles, and has the capacity of detecting new types of intrusions. However, there lies a major difficulty in discovering boundaries between normal and abnormal behavior. This is because there is a deficiency of abnormal samples in the training phase. Also, dynamic anomaly detection [17] requires the system to adapt constantly to normal behavior. Continuous or periodic intrusion detection depends on the way the tool performs its analysis. A static IDS takes snapshots at periodic intervals and analyzes them, looking for vulnerability, configuration errors, etc. This analysis is valid at that precise moment only, and thus does not fulfill the timeliness and performance requirements of an intrusion detection system. A dynamic IDS [4] performs real-time analysis by acquiring information about the environment and the actions taken on the environment at that moment. Monitoring may occur in real-time or in batch. However, this is a costly process, both for transporting and processing data.

2.2. Generic Architecture of Intrusion Detection System

Since most of the intrusions can be uncovered by examining patterns of user activities, many IDSs have been built by utilizing the recognized attack and misuse patterns that can classify a user's activity as normal or abnormal (attack). Data mining techniques are used to analyze audit trails and extract knowledge (features) that help in distinguishing intrusive patterns from normal patterns. [19]

Event Boxes are sensors responsible for data collection and are thus the information sources of an IDS. They can be hardware or software sensors. This information is drawn from various sources, such as network packets, log files and system call traces. The event boxes forward the data collected to the analyzer to determine whether an intrusion has occurred or not. [21]An *analysis box* is used to analyze events and detect hostile behavior. It receives input from the sensors in event boxes and decides whether an intrusion has occurred or not. They can also provide evidence to support their conclusions. the results are sent back to the system, if the IDS uses an active approach.The *database box* stores the event produced by event or analysis boxes, allowing postmortem analysis and guaranteeing persistence.

Response boxes execute a response, to thwart the menace, if an intrusion is detected. An intrusion response is defined as a process which counteracts the effects of an intrusion. In the early stages of a potential attack, the response box may raise an alert in the system.

2.3. Performance Measures of IDS:

The efficiency of intrusion detection systems is evaluated using the following performance measures:

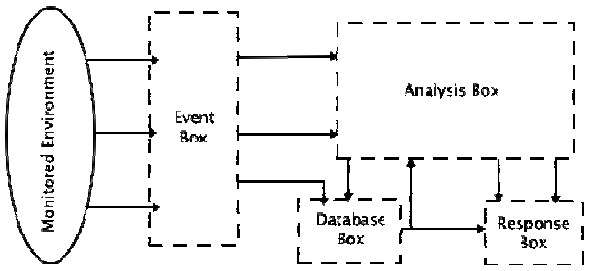

Fig.2 Generic representation of IDS[1]

Accuracy: Inaccuracy occurs when a legitimate action in the environment is flagged as anomalous or intrusive by the intrusion detection system.

Performance: The rate at which audit events are processed is called the performance of the intrusion detection system. Real-time detection is not possible if the performance of the intrusion detection system is poor.

Completeness: Incompleteness occurs when the system fails to detect an attack. This measure is difficult to evaluate than the others, as it is impossible to have global knowledge about attacks and abuses of privileges.

Fault Tolerance: The intrusion detection system should be resistant to attacks and must be designed with this goal in mind.

Timeliness: The analysis by the system must be propagated quickly to enable the security officer to react before much damage has been done, and also to prevent the attacker from subverting the system itself.

3. Related Work

AndroIDS[5] is an open source network-based intrusion detection and prevention system for Android smartphones which was discussed in DEFCON 21. It is a signature based IDS that has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. The system features: Protocol analysis, Content Searching and Content Matching. The IDS architecture primarily consists of a Sensor and a Server. The Sensor runs on the host i.e. the Android smartphone without any human supervision. It analyzes the traffic and sends alerts to warn the user in the event of a threat and reports to the logging Server to drop specific packets. The server receives all the messages that the Android Sensor sends. It sends signatures to remote devices, stores events in databases and detects statistical anomalies in real time.The application uses custom signatures for detecting malicious traffic. The system also derives useful signatures from Snort and emerging threats. The IDS can mitigate attacks like USSD exploits, Webkit vulnerabilities and targeted malwares. The IDS being network-based, would impose restrictions on the mobility of devices.

MADAM [8] is a Multi-Level Anomaly Detector for Android Malware that simultaneously monitors Android at the kernel-level and the user-level to detect real malware infections using machine learning techniques. It is able to distinguish between standard behaviors and malicious ones. Classifiers automatically learn how to classify a set of behaviors. The classifier learns how to do this in the training phase. In order to build a good dataset, one that represents a typical smartphone behavior MADAM considers elements from both, when the phone is active and idle. MADAM relies on multiple computational

intelligence techniques such as clustering, decision trees and probability based classifiers. The learning phase, which follows the training phase has been used to obtain an estimate of the False Positive Rate trend. To reduce the occurrence of false positives, MADAM has to learn how the user behaves during an initialization phase because in this phase the false positives are added to the classifies knowledge base. MADAM has been tested with real Android malware hidden in trojanized applications and has been able to achieve a detection rate of 93%. Even though this IDS tries to address the issue of false positives it is always a lingering problem since newer behavior learnt can still trigger a false positive.

Andromaly [7] is a lightweight Host-based IDS for Android-based mobile devices. The basis of intrusion detection process consists of real-time monitoring, collection, preprocessing and analysis of various system metrics such as CPU consumption, number of packets sent and received, battery usage and number of running processes. Each alert is matched against a set of automatic or manual actions that can be undertaken to mitigate the threat. Automatic actions include among others: uninstalling an application, killing of an active process, changing firewall settings etc. It uses three feature selection methods for the datasets: Information Gain(IG), Chi Square(CS), Fisher Score(FS).

For experiments, 23 games and 20 tools were collected. 11 of these applications were available in Android framework and rest was taken from the Android market. Total 88 features were collected for each monitored application. The representative feature vectors in the training set and the real class of each vector (as game/tool) are assumed to be known and fit to calibrate the detection algorithms (such as decision trees, or Bayesian network). By processing these vectors, the algorithm generates a trained classifier. Although the system is successful in differentiating between the games and the tools applications using machine learning techniques, the classifiers do not give the indistinguishable performance on various devices and hence the application specific activities could not be isolated.

SwarmDroid [6] is a swarm optimized Android IDS developed to address vulnerabilities such as GSM based pivot attacks, mobiles botnets and malicious applications. It treats malware detection as a binary classification problem, where an Android application package (APK) file can contain either malware or goodware. Features were retrieved from each APK file and used for malware detection. Some of these features were redundant and Particle Swarm Optimization (PSO) is used for optimal feature selection, thus reducing computational burdens of the IDS and improving classification accuracy of the Support Vector Machine (SVM), which is used for binary classification of the selected features. The optimization problem involves searching the solution space of the feature set according to a specified criterion for an optimal or near-optimal subset of features. The IDS is tested using a publicly available dataset NSL-KDD. It consists of selected records of the KDD Cup '99 data set and does not have an uneven distribution of attacks, like KDD.The performance was measured using detection time, true positive rate, false positive rate and detection accuracy as evaluation metrics. As a result of optimal feature subset selection using PSO, the malware detection efficiency of SwarmDroid surpassed that of conventional SVM. However the system does not perform real time monitoring of threats.

A comparison of the studied systems is presented in Table 1.

Table 1: Comparison of Intrusion Detection Systems

System	**Technique/Algorithms Used**	**Detection Done**
AndroIDS	Open Source NIDS and IPS with protocol analysis, content searching and content matching.	USSD exploits, Webkit vulnerabilities, targeted Android malware.
Andromaly	HIDS, Real Time. Tested using the following algorithms: k-Means, Logistic Regression, Histograms, Decision Trees, Bayesian Networks, Naive Bayes.	Experiments in the testing phase indicate that it could classify between game/tool features.
MADAM: Multi-level Anomaly Detector for Android Malware	Host-based, real-time anomaly detector. Uses K-Nearest neighbor classifier among other unspecified classifiers.	Monitors Android at both kernel-level and user-level to detect real time threats.

SwarmDroid	Based on the following 4 algorithms: 1) Support Vector Machine (SVM) 2) Particle Swarm Optimization 3) J48 Decision Trees 4) Random Forest Algorithm.	This IDS model design treats malware detection as a binary classification problem with Android APK files containing either mal- or goodware.

4. Types of Attacks

Active attacks: An active attack is any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset. This can be done through viruses, worms or Trojan horses. DoS, MITM, SQL/Javascript injection, data modification are well known active attacks.

Denial of Service Attack (DoS): It is an attack to make a machine or resource unavailable to legitimate users by making some computing or memory resource too busy to handle legitimate requests. [9]

Distributed attacks: In a typical distributed attack, the assailant begins by exploiting vulnerability in one computer system and making it the attack master. This botmaster identifies other vulnerable systems and infects them with malware.

Hijack attacks: It is when the attacker gains unauthorized access to the user sessions over a protected network. There are four ways to compass session hijack attacks: session fixation, session sidejacking, session key theft and cross-site scripting. IP spoofing and man-in-the-middle are well known session attacks.

Malwares: Malware, short for malicious software, is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. It can appear in the form of executable code, scripts, active content and other software.

Passive attacks: Monitor unencrypted data and clear-text passwords that can be used in other types of attacks. Eavesdropping is the most common form of passive attacks. [10]

Probing Attack: It is a way of gathering information about a network of computers in order to circumvent its security controls.

Remote to Local Attack: This occurs when an attacker who is unauthorized to access an account on a machine but can send packets on a network, gains local access as a user of that machine by exploiting some vulnerability.

User to Root Attack: In this type of attack the attacker gains root access to the system by exploiting some vulnerability in the system, starting out with access to a normal user account on the system.

USSD exploits: Unstructured Supplementary Service Data (USSD) is a protocol used by GSM cellular telephones to communicate with the service provider's computers. They are mainly used by network operators for services like call forwarding, balance inquires and multiple SIM functions.
*WebKit Vulnerabilities:*WebKit is a layout engine software component for rendering web pages in web browsers.

5. Open Issues

Intrusion detection systems face some issues when it comes to security and performance. First, the intrusion detection must be more effective, detecting a wider range of attacks and generating fewer false positives. Second, intrusion detection systems must adapt according to networks' increased size, speed and dynamics. There is a trade-off between the security of the system and its performance.

5.1. System Effectiveness

The challenge for system effectiveness is designing an intrusion detection system that detects 100 percent of attacks, while minimizing the number of false positives. In most of the systems studied, intrusion detection relied on misuse detection techniques. Snort [12] (www.snort.org) is an example of this. This requires maintenance of signature sets of all known attacks to analyze network traffic. The developers have to update their signature sets frequently, making this an inefficient approach. Thus, many researchers suggest using a hybrid misuse-anomaly detection approach, but further research is required. [10]

5.2. Performance

It is not only important for IDS to be able to catch attacks but it must also show that it can maintain performance under a variety of conditions. Under increasing network throughput it is assumed that performance of Intrusion

Detection Systems will decrease. The reason for this assumption being that a system would have to carry out packet analysis at a faster rate. This would then result in more intrusions being missed.

5.3. Trade-off between Security and Performance

Integration of high availability and security offers the opportunity for more reliability than systems that solve the problems separately, are easier to implement and offer increased opportunity for recording and analyzing of data. While integrating the two technologies, a fine balance must be achieved between speed and robustness of IDS features. It is equally bad to have the system crash because an attack was missed, or drop large amounts of traffic due to a bottleneck developed in the IDS.

6. Conclusion

In this paper, we have discussed the basics of intrusion detection and prevention systems, taking into account the different types and the performance measures. A generic architecture of IDS has also been provided. A brief description of the types of attacks has been stated. We also present a review of recent works on the different approaches of IDS for smartphones.Finally, the open issues related to the implementation of IDS have been accounted.

References

[1] DimitriosDamopoulos, "Intrusion Detection and Prevention Systems for Mobile Devices: Design and Development,"Ph.D. Thesis, Dept. of Information and Communication Systems Engineering, University of the Aegean, Greece, 2013

[2] McHugh, John, Alan Christie, and Julia Allen. "The role of intrusion detection systems." *Washington Post* (2000)

[3] V. Jyothsna,V. V. Rama Prasad,"A Review of Anomaly based Intrusion Detection Systems." International Journal of Computer Applications (0975 – 8887)Volume 28– No.7, September 2011

[4] Herve Debar, Marc Dacier, Andreas Wespi, "Towards a taxonomy of intrusion-detection systems," in *Computer Networks,* vol. 31, Elsevier, 1999, pp. 802-822

[5] Sanchez, Jaime. Building an Android IDS on Network Level. DEFCON 21, 2013

[6] Adigun, AbimbolaAdebisi, Temitayo Matthew Fagbola, and AdekanmiAdegun. "SwarmDroid: Swarm Optimized Intrusion Detection System for the Android Mobile Enterprise." *International Journal of Computer Science Issues (IJCSI)*11, no. 3 (2014).

[7] AsafShabtai, Uri Kanonov, Yuval Elovici, ChananGlezer, and Yael Weiss. "Andromaly: a behavioral malware detection framework for android devices". Journal of Intelligent Information Systems, pages 1–30, 2011. 10.1007/s10844-010-0148-x.

[8] Dini, Gianluca, Fabio Martinelli, Andrea Saracino, and Daniele Sgandurra. "Madam: a multi-level anomaly detector for android malware." In *Computer Network Security*, pp. 240-253.Springer Berlin Heidelberg, 2012.

[9] Tavallaee, Mahbod, EbrahimBagheri, Wei Lu, and Ali-A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*. 2009.

[10] Matthews, Melantha (2011). "Network Security Attack: Active/Passive Comparison" [Online] Available: http://www.brighthub.com/computing/smb-security/articles/104551.aspx

[11] Kemmerer, Richard A., and Giovanni Vigna. "Intrusion detection: A brief history and overview (supplement to computer magazine)." *Computer* 35.4 (2002): 27-30.

[12] Gordeev, Mikhail (2000). "Intrusion Detection: Techniques and Approaches," [Online] Available: http://www.forum-intrusion.com/archive/Intrusion%20Detection%20Techniques%20and%20Approaches.html

[13] Roesch, Martin. "Snort: Lightweight Intrusion Detection for Networks." *LISA*. Vol. 99. 1999.

[14] S. Axelsson. "Intrusion detection systems: A survey and taxonomy." Technical Report 99-15, Chalmers Univ., March 2000.

[15] Fangfang Yuan, LidongZhai, Yanan Cao and Li Guo, "Research of Intrusion Detection System on Android", in *IEEE Ninth World Congress on Services*, 2013

[16] International Data Corporation. (2014). *Worldwide Smartphone Shipments Q2 2014* [Online] Available: http://www.idc.com/getdoc.jsp?containerId=prUS25037214

[17] Wu, Shelly Xiaonan, and Wolfgang Banzhaf. "The use of computational intelligence in intrusion detection systems: A review." *Applied Soft Computing* 10, no. 1 (2010): 1-35.

[18] Sherif, Joseph S., and Tommy G. Dearmond. "Intrusion detection: systems and models." In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 115-115. IEEE Computer Society, 2002.

[19] Mukkamala, S., A. H. Sung, and A. Abraham. "Designing Intrusion Detection Systems: Architectures." *Challenges and Perspectives* (2003).

[20] Gupta, Kapil Kumar. "Robust and efficient intrusion detection systems.", Ph.D. Thesis, Department of Computer Science and Software Engineering, University of Melbourne, 2009.

[21] Kou, Xiaoming, and Qiaoyan Wen. "Intrusion detection model based on android." In *Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE International Conference on*, pp. 624-628. IEEE, 2011.

Chani Jindal is currently pursuing her B.E.(Computer Engineering) degree from MIT College of Engineering, India.

Her research interests include data mining and networking.

Mukti Chowkwale is currently pursuing her B.E.(Computer Engineering) degree from MIT College of Engineering, India. Her research interests include machine learning, data mining and information security.

Rohan Shethia is currently pursuing his B.E.(Computer Engineering) degree from MIT College of Engineering, India. His research interests include information security and networking.

Sohail Ahmed Shaikhis currently pursuing his B.E.(Computer Engineering) degree from MIT College of Engineering, India. His research interests include information security, networking, artificial intelligence and nanotechnology.